

PATENT

**Attorney Docket No. 678-629 [P9648/ST2 (2000-57330)]
IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re application of: **SUNG-CHUL HAN**

Serial No.: **Unassigned**

Filed: **August 30, 2001**

For: **INTERLEAVER FOR A TURBO ENCODER IN
AN UMTS AND METHOD FOR PERFORMING
INTERLEAVING**



BOX PATENT APPLICATION

**Assistant Commissioner for Patents
Washington, D. C. 20231**

**TRANSMITTAL OF CERTIFIED COPY
OF PRIORITY APPLICATION**

Attached please find the certified copy of the foreign application from which
priority is claimed for this case:

Country: **Korea**
Application No.: **57330/2000**
Filing Date: **September 29, 2000**

Reg. No. **33,494**



SIGNATURE OF ATTORNEY

Tel. No. **(516) 228-8484**

Paul J. Farrell

Type or print name of attorney

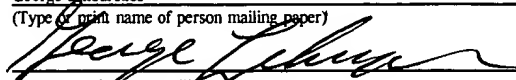
P.O. Address **DILWORTH & BARRESE
333 Earle Ovington Boulevard
Uniondale, New York 11553**

NOTE: The claim to priority need be in no special form and may be made by the attorney or agent if the foreign
application is referred to in the oath or declaration as required by § 1.63.

CERTIFICATION UNDER 37 C.F.R. § 1.10

I hereby certify that this New Application Transmittal and the documents referred to as enclosed therein are being deposited with the United States Postal Service
on this date August 31, 2001 in an envelope as "Express Mail Post Office to Addressee" Mail Label Number EL 814600655 US addressed to: BOX PATENT
APPLICATION, Assistant Commissioner for Patents, Washington, D.C. 20231.

George Likourezos
(Type or print name of person mailing paper)



(Signature of person mailing paper)

P9648

11000 U.S. PTO
09/943895
08/31/01

96

대한민국 특허청
KOREAN INTELLECTUAL
PROPERTY OFFICE

별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto
is a true copy from the records of the Korean Intellectual
Property Office.

출원번호 :
Application Number

특허출원 2000년 제 57330 호

출원년월일 :
Date of Application

2000년 09월 29일

출원인 :
Applicant(s)

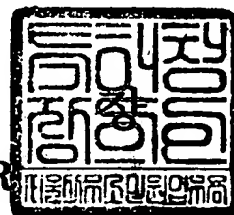
삼성전자 주식회사



2001 년 03 월 07 일

특 허 청

COMMISSIONER



【서류명】	특허출원서		
【권리구분】	특허		
【수신처】	특허청장		
【참조번호】	0009		
【제출일자】	2000.09.29		
【국제특허분류】	H04M		
【발명의 명칭】	유 . 엠 . 티 . 에스시스템내 터보부호화기의 내부 인터리버 및 인터리빙 수행 방법		
【발명의 영문명칭】	AN INTERNAL INTERLEAVER OF THE TURBO DECODER IN AN UMT SYSTEM AND METHOD FOR INTERLEAVING THEREOF		
【출원인】			
【명칭】	삼성전자 주식회사		
【출원인코드】	1-1998-104271-3		
【대리인】			
【성명】	이건주		
【대리인코드】	9-1998-000339-8		
【포괄위임등록번호】	1999-006038-0		
【발명자】			
【성명의 국문표기】	한성철		
【성명의 영문표기】	HAN, Sung Chul		
【주민등록번호】	710706-1841120		
【우편번호】	463-050		
【주소】	경기도 성남시 분당구 서현동 시범단지 한양아파트 319-1705		
【국적】	KR		
【심사청구】	청구		
【취지】	특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사를 청구합니다. 대리인 이건주 (인)		
【수수료】			
【기본출원료】	20	면	29,000 원
【가산출원료】	8	면	8,000 원
【우선권주장료】	0	건	0 원
【심사청구료】	9	항	397,000 원
【합계】	434,000	원	

【첨부서류】

1. 요약서·명세서(도면)_1통

【요약서】**【요약】**

본 발명은 UMTS 시스템의 터보 부호화기내 인터리버에 있어서, 인터리버 동작 조건을 설정하기 위한 레지스터부와, 상기 레지스터부로부터 조건을 입력받아 시스템 동작에 필요한 제어 신호를 생성하는 제어부와, 상기 제어부에서 생성된 제어 신호에 따라 상기 레지스터부에서 제공되는 행간치환 패턴, 행내치환 패턴 증가분 배열, 행내치환용 기본수열을 이용하여 최종 인터리빙된 주소를 생성하는 주소 계산부, 상기 터보 부호화기로 입력되는 입력 데이터를 순차적으로 저장해 두었다가 상기 주소 계산부에서 생성된 주소에 해당하는 데이터를 출력하는 데이터 저장부로 구성함을 특징으로 한다. 특히 상기 주소 계산부는 행내치환 패턴 생성부와, 상기 행내치환 패턴 생성부가 행내치환 패턴 계산시에 중간 데이터를 저장하는 행내치환 패턴 저장 배열부, 상기 제어부의 제어 신호에 따라 레지스터부로부터 행간치환 패턴 해당값과 행내치환용 기본수열의 해당값을 취하여 인터리빙된 주소를 계산하는 출력 주소 생성부를 포함하여 구성함을 특징으로 한다.

【대표도】

도 2

【색인어】

터보부호화기, UMTS, 인터리버, 행/열변환

【명세서】**【발명의 명칭】**

유.엠.티.에스시스템내 터보부호화기의 내부 인터리버 및 인터리빙 수행 방법{AN INTERNAL INTERLEAVER OF THE TURBO DECODER IN AN UMTS SYSTEM AND METHOD FOR INTERLEAVING THEREOF}

【도면의 간단한 설명】

도 1은 종래 터보 부호화기의 블록 구성도,

도 2는 본 발명의 실시 예에 따른 터보 부호화기의 내부 인터리버 블록 구성도,

도 3은 본 발명의 실시 예에 따른 행내치환 패턴[a(j)] 생성부의 상세 회로 구성도,

【발명의 상세한 설명】**【발명의 목적】****【발명이 속하는 기술분야 및 그 분야의 종래기술】**

<4> 본 발명은 무선 통신시스템의 터보 부호화기에 관한 것으로, 특히 UMTS(Universal Mobile Telecommunication System) 시스템내 터보 부호화기의 내부 인터리버 및 인터리빙 수행 방법에 관한 것이다.

<5> 일반적으로 위성시스템, ISDN(Integrated Service Digital Network), 디지털 셀룰라, W-CDMA(Code Division Multiple Access), IMT-2000(CDMA 2000) 등과 같은 무선 통신

시스템의 부호기로는 터보부호(Turbo code)를 사용하는 부호화기가 사용될 수 있다. 이러한 터보 부호화기의 내부에는 인터리버가 포함되는데, 이 인터리버는 부호화기에 입력되는 데이터를 랜덤화함으로써 코드워드의 디스턴스 특성을 개선시키는 것으로 터보 부호화기의 성능을 좌우하는 중요한 요소이다.

<6> 도 1은 통상적인 터보 부호화기의 블록 구성을 도시한 것으로, 상기 도 1에 도시된 바와 같이 터보 부호화기(106)는 입력 프레임 데이터 dk를 부호화하여 Y1k로 출력하는 제1구성 부호화기(constituent encoder)(102)와, 상기 입력 프레임 데이터 dk를 인터리빙하는 인터리버(interleaver)(100)와, 상기 인터리버(100)의 출력을 부호화하여 Y2k로 출력하는 제2구성 부호화기(104)를 구성된다. 즉, 터보 부호화기(106)는 입력 프레임 데이터 dk를 부호화 처리하지 않고 바로 Xk로 출력할 뿐만 아니라 부호화 처리하여 Y1k를 출력하고, 인터리빙 및 부호화 처리하여 Y2k를 출력한다. 이중 상기 인터리버(100)는 입력되는 데이터의 프레임 크기와 동일한 크기의 데이터를 출력하며, 상기 제2구성 부호화기(104)로 입력되는 데이터 비트의 입력 순서를 변경시킴으로써 데이터 비트들 사이의 상관을 줄여 데이터의 보안을 증가시킨다.

<7> 이하 상기 인터리버(100)의 동작을 좀더 자세히 살펴보면,

<8> 먼저 인터리버(100)는 입력되는 데이터를 행(Row)과 열(Column)로 구성된 메모리에 순차적으로 저장시킨 후 입력되는 데이터의 수에 따라 정해지는 인터리빙 변수값을 초기화 한다. 즉, 인터리버(100)는 상기 입력 데이터 수 K에 따른 행 수 R과 열 수 C 뿐만 아니라 입력 데이터의 인터리빙 수행을 위한 행간/행내치환을 위한 소수 p와 μ 등과 같은 각종 변수값을 메모리내 테이블을 참조하여 상기 입력 데이터 수에 대응되는 값으로 설정한다.

<9> 즉, 만일 상기 입력데이터 수가 $K = 40$ 이라고 한다면, 상기 각각의 변수는 각각 $R = 5$, $C = 8$, $p = 7$, $\mu = 3$ 등으로 정해지게 된다. 그러면 인터리버(100)는 상기 입력데이터를 하기의 [표 1]에서 보여지는 바와 같은 5×8 행렬로 메모리상에 위치 시킨다.

<10> 【표 1】

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40

<11> 이어 인터리버(100)는 상기 [표 1]에서와 같이 위치된 입력 데이터의 행과 열중 먼저, 행에 대해 인터리빙 동작을 수행한다. 이때 인터리버(100)는 상기 행에 대한 인터리빙 동작을 위해 행간치환(inter-row) 패턴[$T(j)$]를 결정하게 된다. 상기 치환패턴이란 함은 입력 데이터의 행간 인터리빙을 위한 함수로써 상기 입력데이터 수 K 에 따라 아래의 [수학식 1]에서와 같이 결정된다.

<12> 【수학식 1】

$$T(j) = \text{pat4} \quad (40 \leq k \leq 159)$$

<13> $= \text{pat3} \quad (160 \leq k \leq 200)$

<14> $= \text{pat1} \quad (201 \leq k \leq 480)$

<15> $= \text{pat3} \quad (481 \leq k \leq 530)$

<16> $= \text{pat1} \quad (531 \leq k \leq 2280)$

<17> $= \text{pat2} \quad (2281 \leq k \leq 2480)$

<18> =pat1 ($2481 \leq k \leq 3160$)

<19> =pat2 ($3161 \leq k \leq 3210$)

<20> =pat1 ($3211 \leq k \leq 5114$)

<21> 단, pat1= {19,9,14,4,0,2,5,7,12,18,10,8,13,17,3,1,16,6,15,11}

<22> pat2= {19,9,14,4,0,2,5,7,12,18,16,13,17,15,3,1,6,11,8,10}

<23> pat3= {9,8,7,6,5,4,3,2,1,0}

<24> pat4= {4,3,2,1,0}

<25> 이어 인터리버(100)는 상기 T(j)치환 패턴값에 따라 원본 행렬의 T(j)번째 행을 새로운 행렬의 j번째에 복사하여 상기 입력 데이터의 행을 변경시키게 된다. 즉, 입력 데이터 수 K가 '40'임에 따라 T(j)값은 pat4가 선택되며, 상기 선택된 치환 수열값에 따라 행변환된 입력 데이터의 행렬을 도시한 하기 [표 2]에서와 같이, 상기 원본 행렬의 4행의 데이터는 새로운 행렬의 0번째 행으로, 3행의 데이터는 새로운 행렬의 1번째 행으로, 2행의 데이터는 새로운 행렬의 2번째 행으로, 1행의 데이터는 새로운 행렬의 3번째 행으로, 0행의 데이터는 새로운 행렬의 4번째 행으로 각각 위치시키게 된다.

<26> 【표 2】

33	34	35	36	37	38	39	40
25	26	27	28	29	30	31	32
17	18	19	20	21	22	23	24
9	10	11	12	13	14	15	16
1	2	3	4	5	6	7	8

<27> 이어 인터리버(100)는 행내치환을 위한 치환수열들인 최소 소수 수열 $[q(j)]$, 치환 후 소수 수열 $[r(j)]$, 행내치환용 기본수열 $[s(i)]$ 를 아래의 [수학식 2]에 따라 차례로 구하게 된다.

<28> 【수학식 2】

$$g.c.d\{q(j), p-1\} = 1, \quad q(j) > 6, \quad q(j) > q(j-1),$$

$$q=1, 2, \dots, R-1, \quad q(0)=1$$

<29> (단, $g.c.d$ 는 최대공약수)

<30> $r[T(j)] = q(j), \quad j = 0, 1, \dots, R-1.$

<31> $s(i) = [\mu \times (i-1)] \bmod p, \quad i = 1, 2, \dots, (p-2), \quad s(0)=1$

<32> 그리고 인터리버(100)는 행간/행내치환을 위한 최종 행내치환 패턴 $U^j(i)$ 를 상기 행내치환용 기본수열 $[s(i)]$ 을 이용하여 아래의 [수학식 3]에서와 같이 구하게 된다.

<33> 【수학식 3】

$$1. \quad U^j(i) = s\{[i \times (j)] \bmod (p-1)\}, \quad i=0, 1, 2, \dots, p-2, \quad U(p-1)=0$$

($C=p$ 인 경우)

<34> $2. \quad U^j(i) = s\{[i \times (j)] \bmod (p-1)\}, \quad i=0, 1, 2, \dots, p-2, \quad U(p-1)=0, \quad U(p)=p$

<35> ($C=p+1$ 인 경우) 단, $K=C \times R$ 인 경우 행내 치환 후, $U(p)$ 와 $U(0)$ 을 교환

<36> $3. \quad U^j(i) = s\{[i \times (j)] \bmod (p-1)\}-1, \quad i=0, 1, 2, \dots, p-2, \quad (C=p \text{ 인 경우})$

<37> 즉, 상기 최종 행내치환 패턴 $U^j(i)$ 값은 각 열에 대해 아래의 [수학식 4]에서와 같

이 구해지게 된다.

<38> 【수학식 4】

$$U^0(i) = s(17i \bmod 6) = s\{0, 5, 4, 3, 2, 1, 0, 7\} = \\ \{1, 5, 4, 6, 2, 3, 0, 7\}$$

<39> $U^1(i) = s(13i \bmod 6) = s\{0, 1, 2, 3, 4, 5, 0, 7\} = \{1, 3, 2, 6, 4, 5, 0, 7\}$

<40> $U^2(i) = s(11i \bmod 6) = s\{0, 5, 4, 3, 2, 1, 0, 7\} = \{1, 5, 4, 6, 2, 3, 0, 7\}$

<41> $U^3(i) = s(7i \bmod 6) = s\{0, 1, 2, 3, 4, 5, 0, 7\} = \{1, 3, 2, 6, 4, 5, 0, 7\}$

<42> $U^4(i) = s(i \bmod 6) = s\{0, 1, 2, 3, 4, 5, 0, 7\} = \{7, 3, 2, 6, 4, 5, 0, 1\}$

<43> 따라서 인터리버(100)는 상기 행내치환을 위한 최종 행내치환 패턴 $U^j(i)$ 에 따라
상기 [표 2]에서와 같이 행간치환된 데이터 열을 아래의 [표 4]에서와 같이 행내치환
(intra-row)하여 최종 인터리빙을 수행하게 된다.

<44> 【표 3】

$$\begin{bmatrix} 34 & 36 & 37 & 34 & 35 & 36 & 33 & 40 \\ 26 & 28 & 27 & 31 & 29 & 30 & 25 & 32 \\ 18 & 22 & 21 & 23 & 19 & 20 & 17 & 24 \\ 10 & 12 & 11 & 15 & 13 & 14 & 9 & 16 \\ 8 & 4 & 3 & 7 & 5 & 6 & 1 & 2 \end{bmatrix}$$

<45> 이상에서 살펴본 바와 같이 종래 터보 부호화기의 인터리버는 입력 데이터의 인터
리빙을 위해 상기한 바와 같이 치환된 데이터의 중간 저장을 위해 많은 메모리를 요구하

고 마이크로 프로세서 및 DSP에 과부하를 초래하게 되는 문제점이 있었다.

【발명이 이루고자 하는 기술적 과제】

<46> 따라서 본 발명의 목적은 UMTS 터보부호화기내 인터리버에서 인터리빙을 위한 데이터 처리로 인한 마이크로 프로세서의 과부하를 방지하고 중간 저장에 필요한 메모리를 제거할 수 있도록 하는 인터리버 및 인터리빙 수행 방법을 제공함에 있다.

<47> 상술한 목적을 달성하기 위한 본 발명은 본 발명은 UMTS 시스템의 터보 부호화기내 인터리버에 있어서, 인터리버 동작 조건을 설정하기 위한 레지스터부와, 상기 레지스터부로부터 조건을 입력받아 시스템 동작에 필요한 제어 신호를 생성하는 제어부와, 상기 제어부에서 생성된 제어 신호에 따라 상기 레지스터부에서 제공되는 행간치환 패턴, 행내치환 패턴 증가분 배열, 행내치환용 기본수열을 이용하여 최종 인터리빙된 주소를 생성하는 주소 계산부, 상기 터보 부호화기로 입력되는 입력 데이터를 순차적으로 저장해 두었다가 상기 주소 계산부에서 생성된 주소에 해당하는 데이터를 출력하는 데이터 저장부로 구성함을 특징으로 한다. 특히 상기 주소 계산부는 행내치환 패턴 생성부와, 상기 행내치환 패턴 생성부가 행내치환 패턴 계산시에 중간 데이터를 저장하는 행내치환 패턴 저장 배열부, 상기 제어부의 제어 신호에 따라 레지스터부로부터 행간치환 패턴 해당값과 행내치환용 기본수열의 해당값을 취하여 인터리빙된 주소를 계산하는 출력 주소 생성부를 포함하여 구성함을 특징으로 한다.

【발명의 구성 및 작용】

<48> 이하 본 발명의 바람직한 실시 예를 첨부한 도면을 참조하여 상세히 설명한다. 상기 첨부 도면의 구성 요소들에 참조부호를 부가함에 있어서는 동일한 구성요소들에 한해서는 비록 다른 도면상에 표시되더라도 가능한 한 동일한 부호를 가지도록 하고 있음에 유의하여야 한다. 또한 하기 설명 및 첨부 도면에서 구체적인 처리 흐름과 같은 많은 특정 상세들이 본 발명의 보다 전반적인 이해를 제공하기 위해 나타나 있다. 이들 특정 상세들없이 본 발명이 실시될 수 있다는 것은 이 기술분야에서 통상의 지식을 가진 자에게 자명할 것이다. 그리고 본 발명의 요지를 불필요하게 흐릴 수 있는 공지 기능 및 구성에 대한 상세한 설명은 생략한다.

<49> 도 2는 본 발명의 실시 예에 따른 터보부호화기내 인터리버의 블록 구성을 도시한 것이다. 상기 도 2를 참조하면, 먼저 입력 데이터 수 K 및 이에 따라 결정되는 상수 R , C , p , μ , TypeD가 레지스터부(200)에 저장된다. 이 중 TypeD는 $C=p+1$ 이고 $K=C \cdot R$ 일 때 1로 설정한다.

<50> 또한 행간치환 패턴[$T(j)$], 행내치환용 기본수열[$s(i)$], 행내치환 패턴 증가분 배열[$incr(j)$]이 레지스터부(200)내의 메모리에 저장된다. 상기 행간치환 패턴[$T(j)$], 행내치환용 기본수열[$s(i)$], 행내치환 패턴 증가분 배열[$incr(j)$]은 제어부(도시하지 않았음)로부터 i 및 j 값을 입력받아 그에 해당하는 값을 주소 계산부(202)에 전달한다. 이때 상기 행간치환 패턴[$T(j)$]은 상기 [수학식 1]과 같이 결정되며, 행내치환용 기본수열[$s(i)$]은 상기 [수학식 2]에서와 같이 결정된다. 그리고 행내치환 패턴 증가분 배열[$incr(j)$]은 아래의 [수학식 5]에서와 같이 결정된다.

<51> 【수학식 5】

$$\text{incr}(j)=r(j) \bmod (p-1), j=0, 1, \dots, R-1$$

<52> 상기 [수학식 5]에서 사용되는 치환후 소수 수열[r(j)]를 구하는 과정에서 본 발명의 실시 예에서는 전술한 [수학식 2]에서와 같이 행간치환 패턴[T(j)]를 이용하는 대신 이를 역변환한 행간 역치환 패턴[TI(j)]를 이용함으로써, 치환후 소수 수열[r(j)]의 계산에 걸리는 연산량을 줄일 수 있다. 이를 아래의 [수학식 6]에 보였다.

<53> 【수학식 6】

$$r(j)=q[\text{TI}(j)], j=0, 1, \dots, R-1$$

<54> 여기서 상기 최소 소수 수열[q(j)]는 상기 [수학식 2]에서와 동일하며, TI(j)는 행간 역치환 패턴으로서 아래의 [수학식 7]에서와 같이 결정된다.

<55> 【수학식 7】

$$\text{TI}(j)=\text{pat6} \quad (2281 \leq K \leq 2480 \text{ 일 때})$$

$$\text{TI}(j)=\text{pat5} \quad (2481 \leq K \leq 3160 \text{ 일 때})$$

$$\text{TI}(j)=\text{pat6} \quad (3161 \leq K \leq 3210 \text{ 일 때})$$

$$\text{TI}(j)=\text{pat5} \quad (3211 \leq K \leq 5114 \text{ 일 때})$$

$$\text{단, pat5}=\{4, 15, 5, 14, 3, 6, 17, 7, 11, 1, 10, 19, 8, 12, 2, 18, 16, 13, 9, 0\}$$

$$\text{pat6}=\{4, 15, 5, 14, 3, 6, 16, 7, 18, 1, 19, 17, 8, 11, 2, 13, 10, 12, 9, 0\}$$

$$\text{pat7}=\{9, 8, 7, 6, 5, 4, 3, 2, 1, 0\}$$

<62> pat8= {4,3,2,1,0}

<63> 즉, 레지스터부(200) 설정이 완료된 후 하드웨어부 내부의 제어부는 주소 계산부 (202)의 동작을 제어하기 위한 카운터값 i, j 를 순차적으로 생성한다. 여기서 i, j 는 각각 0에서 시작하며 j 가 1씩 증가해서 $R-1$ 에 도달한 후 i 가 1이 증가한다. 이러한 방식으로 i, j 가 각각 $C-1, R-1$ 에 도달할 때까지 계속된다.

<64> 한편, 상기 [수학식 3]에서 최종 행내 치환 패턴[$U_j(i)$]를 구할 때 행내치환용 기본수열[$s(i)$] 내의 주소를 각 행별 행내치환 패턴[$a^i(j)$]라 정의하면 행내치환 패턴[$a^i(j)$]는 아래의 [수학식 8]에서와 같이 표현될 수 있다.

<65> 【수학식 8】

$$a^i(j) = [i \times r(j)] \bmod (p-1)$$

<66> 단, $U_j(i) = s[a^i(j)]$, $j=0,1,2,\dots,R-1$, $i=0,1,2,\dots,p-2$

<67> 본 발명에서는 상기 [수학식 8]에서 정의된 각 행별 행내치환 패턴[$a^i(j)$]이 j 를 고정시키고 i 를 1씩 증가시킴에 따라 초기값 '0'으로부터 일정한 크기씩 증가된 후 $(p-1)$ 로 나누어진 나머지의 값으로 나타남에 착안하여 그 증가치를 행내치환 패턴 증가분 배열[incr(j)]에 저장시켜 두었다가 실제 하드웨어부의 동작시에는 각 행에 해당하는 행내치환 패턴 증가분을 이용하여, 곱셈을 수행하는 대신 덧셈과 나머지 연산만으로 행내치환 패턴[$a^i(j)$]을 생성한다. 이를 아래의 [수학식 9]에 보였다.

<68> 【수학식 9】

$$a^i(j)=[a^{i-1}(j)+incr(j)] \bmod (p-1),$$

<69> 단, $j=0,1,2,\dots,R-1$, $i=1,2,\dots,p-2$, $a^0(j)=0$

<70> 즉, 상기 [수학식 9]에서 보여지는 바와 같이 각각의 j 에 대해 행내치환 패턴 $[a^i(j)]$ 은 i 가 증가하기 전의 값인 $a^{i-1}(j)$ 로부터 계산될 수 있다. 또한 i 는 인터리빙이 수행되는 동안 감소하지 않으므로 현재값을 계산하고 난 후에는 이전값을 저장해 둘 필요 없다. 이와 같은 계산 방식에 근거하여 행내치환 패턴 생성부 $[a(j)]$ (204)는 각각의 i, j 값에 대해 행내치환 패턴 $[a^{i-1}(j)]$ 을 생성할 때마다 각각의 j 에 대한 마지막 $a^i(j)$ 값을 행내치환 패턴 저장 배열부(206)에 저장해 둔다.

<71> 하드웨어부의 동작시에 행내치환 패턴 생성부 $[a(j)]$ (204)는 제어부로부터 제공되는 각각의 i, j 값에 대해 행내치환 패턴을 생성하는데, i 가 0일 때는 행내치환 패턴 저장 배열부(206)를 초기화시켜야 하므로 행내치환 패턴 $[a(j)]$ 으로 0을 출력한 후 이를 행내치환 패턴 저장 배열부(206)에 저장한다. i 가 0보다 크고 $p-2$ 와 같거나 작을 때는 $a(j)$ 에서 j 번째에 해당하는 데이터를 읽어와서 행내치환 패턴 증가분 배열 $[incr(j)]$ 에서 읽어온 해당 증가분을 더한 후 나머지 연산을 수행하여 새로운 행내치환 패턴을 생성한 후 이를 출력한다. 출력된 데이터는 다음 행내치환 패턴 생성을 위하여 다시 행내치환 패턴 저장 배열부(206)의 j 번째 주소에 저장된다.

<72> 행내치환 패턴 생성부 $[a(j)]$ (204)에서 출력되는 행내치환 패턴 $[a^i(j)]$ 은 레지스터부(200) 내부의 행내치환용 기본 수열 $[s(i)]$ 메모리(208)에 리드 어드레스로 입력되며,

이에 따라 이 메모리(208)의 출력 $s[a(j)]$ 는 최종 주소 생성부(207)에 전달된다.

<73> 최종 주소 생성부(207)는 상기 제어부로부터 제공되는 i, j 값에 따라 행내치환용 기본수열의 해당값인 $s[a(j)]$ 를 취하여 먼저 최종 행내 치환 패턴 [$U^j(i)$]을 계산하고, 이 계산결과와 레지스터부(200)로부터 입력된 행간치환 패턴 [$T(j)$] 해당값을 이용하여 최종 인터리빙된 주소($xaddr$)를 계산한다. 또한 TypeD가 1인 경우의 예외처리를 수행한다. 상기 최종 주소 생성부(207)의 계산 과정을 아래의 [수학식 10]과 [수학식 11]에 보인다.

<74> 【수학식 10】

$$U^j(i) = s[a(j)] \quad (0 \leq i \leq p-2 \text{ 인 경우}),$$

<75> $U^j(i) = 0 \quad (i = p-1 \text{ 인 경우}),$

<76> $U^j(i) = p \quad (i = p \text{ 인 경우}),$

<77> $i = 0, 1, 2, \dots, C-1, \quad j = 0, 1, 2, \dots, R-1$

<78> 단 TypeD가 1인 경우에는 다음과 같은 예외 처리

<79> $U^j(i) = p \quad (i = 0 \text{ 이고 } j = R-1 \text{ 인 경우}),$

<80> $U^j(i) = 1 \quad (i = p \text{ 이고 } j = R-1 \text{ 인 경우})$

<81> 【수학식 11】

$$xaddr = C * T(j) + U^j(i)$$

<82> 데이터 저장부(214)는 최종 주소 생성부(207)에서 생성된 최종 인터리빙된 주소

(xaddr)에 해당하는 데이터를 독출하여 이를 출력한다. 도 3는 상기 행내치환 패턴 $[a(j)]$ 생성부(204)의 상세회로 구성을 도시한 것으로, 상기 도 3를 참조하여 행내치환 패턴 $[a(j)]$ 생성부(204)의 동작을 좀더 자세히 설명하도록 한다.

<83> 먼저 상기 행내치환 패턴 $[a(j)]$ 의 변수 i 가 0인 경우 제2멀티플렉서(402)로부터 행내치환 패턴 $[a(j)]$ 은 초기값으로 0이 선택되어 출력되고, 이는 레지스터부(200) 내의 행내치환용 기본수열 $[s(i)]$ 의 주소로 인가됨과 동시에 행내치환 패턴 저장 배열부(204)의 j 번째 주소에 저장된다. 변수 i 가 '0'이 아닐 경우에는 행내치환 패턴 저장 배열부(206)의 j 번째 주소의 데이터값이 제1가산기(400)로 피드백되어 이전 변수 데이터값으로 입력되는데, 이는 이전에 저장해 둔 행내치환 패턴 $[a^{i-1}(j)]$ 에 해당한다. 이때 상기 제1가산기(400)의 다른 한 입력단으로는 상기 행내치환 패턴 증가분 배열 $[incr(j)]$ 의 j 번째 주소에 저장된 값이 레지스터부(200)로부터 상기 제1가산기(400)로 입력된다. 이에 따라 상기 제1가산기(400)에서는 상기 이전 행내치환 패턴 $[a(j)]$ 값과 증가분 $[incr(j)]$ 값이 가산되며, 상기 제1가산기(400)의 출력값인 제1가산치는 제2가산기(404)로 인가되어 상기 [수학식 5]에서 보여지는 바와 같이 제산 계수 $[-(p-1)]$ 값이 가산된 후 제1멀티플렉서(408)와 부호검출부(406)로 각각 인가된다. 즉, 상기 제2가산기(404)에서는 이전 행내치환 패턴 $[a(j)]$ 값과 증가분 $[incr(j)]$ 값이 가산된 제1가산치를 다시 제산 계수 $(p-1)$ 로 나눈 나머지값 $\{a(j)+incr(j)\}-(p-1)$ 이 출력되게 된다. 이는 상기 [수학식 9]에 보인 바와 같이 제산 계수 $(p-1)$ 로 나머지 연산을 하기 위해서는 상기 이전 행내치환 패턴 $[a(j)]$ 값에 증가분 $[incr(j)]$ 값을 가산한 값이 제산 계수 $(p-1)$ 보다 클 경우 그 제1가산치의 결과에서 $(p-1)$ 를 뺀 값이 나머지에 해당하는 값이 되기 때문이다. 상기 제2가산기(404)로부터 출력된 제2가산치 $\{a(j)+incr(j)\}-(p-1)$ 은 최상위비트 부호 검출부(406)와 제1멀티플

렉서(408)로 각각 입력된다. 그러면 부호 검출부(406)에서는 상기 제2가산기(404)로부터 출력된 제2가산치의 최상위 비트만을 검출하여 이를 제1멀티플렉서(408)의 선택제어신호로 인가한다. 이에 따라 제1멀티플렉서(408)는 상기 부호 검출부(406)로부터의 선택제어신호에 따라 상기 제1가산기(400)로부터 출력되는 제1가산치 $a(j)+incr(j)$ 과 상기 제2가산기(404)로부터 출력되는 제2가산치 $\{a(j)+incr(j)\}-(p-1)$ 중 한 값을 선택하여 출력시킨다. 이 때 부호 검출부(406)에서 검출된 부호가 '0'이면 제1가산치가 $(p-1)$ 과 같거나 커서 $(p-1)$ 을 빼야 나머지 연산의 결과가 됨을 의미하고, 부호가 '1'이면 제1가산치 결과가 $(p-1)$ 보다 작아서 제1가산치의 결과 그대로 나머지 연산의 결과가 됨을 의미한다. 이에 따라 i 값이 초기 0인 이후에 제2멀티플렉서(408)로부터 출력되는 상기 제1 또는 제2가산치는 제1멀티플렉서(402)를 통해 올바른 나머지 연산 결과를 선택하여 새로운 행내치환 패턴 $[a(j)]$ 을 레티스터부(200) 내의 행내치환용 기본수열 $[s(i)]$ 의 주소로 인가함과 동시에 행내치환 패턴 저장 배열부(206)의 j 번째 주소에 저장하는 것이다.

<84> 이하 상기 열변환 변수 데이터열 $[a(j)]$ 생성부(204)의 동작을 입력 데이터의 최종 행내치환 패턴 $[U_j(i)]$ 의 계산식을 도식한 상기 [수학식 4]의 $U^0(i)$ 값 계산식을 예를 들어 살펴보면, 즉 j 가 '0'이고 i 가 0인 경우 행내치환용 기본수열 $[s(i)]$ 참조시의 출력 주소 i 값, 즉, 행내치환 패턴 $[a(j)]$ 값은 0으로 설정된다. 그러면 상기 행내치환 패턴 $[a(j)]$ 의 초기값 0은 행내치환 패턴 저장 배열부(206)에 저장됨과 동시에 행내치환용 기본수열 $[s(i)]$ 의 주소로 인가된다. 즉, 동일한 j 값('0')에 대해 i 가 1로 증가했을 때 행내치환 패턴 저장 배열부(206)에 저장되어 있던 이전 행내치환 패턴값 '0'이 제1가산기(400)로 입력되어 증가분 $[incr(j)]$ 값 5와 가산된다. 그 결과인 제1가산치 5의 값은 제1멀티플렉서(408)의 한 입력단자로 인가됨과 동시에 제2가산기(404)로 인가된다. 그러면

제2가산기(404)에서 상기 제1가산치 5는 제산 계수 $-(p-1)$ 즉, -6와 가산되어 -1의 값으로 부호 검출부(406)로 출력되는데, 이때 부호 검출부(406)는 상기 제2가산치가 음의 값이므로 제1멀티플렉서(408)가 상기 제1가산기(400) 출력인 제1가산치를 선택하도록 하는 선택제어신호 '1'을 발생하여 인가한다. 이에 따라 제1멀티플렉서(408)로부터 제1가산치인 5가 출력되고 이값은 다시 제2멀티플렉서(402)를 통해 다음 행내치환 패턴[a(j)]으로 선택되어 행내치환 패턴 저장 배열부(206)에 저장되게 된다. 즉 이와 같은 방법으로 종래 행내치환용 기본수열[s(i)]의 주소 i값을 상기 [수학식 2]에서와 같이 복잡하게 구하지 않아도 되어 회로의 구성이 간단해지게 되는 것이다.

<85> 한편 상기 레지스터부(200)에는 상기한 증가분[incr(j)]값 외에도 소수를 의미하는 μ , 입력 데이터 수를 의미하는 K, 입력 데이터의 행 수를 의미하는 R, 입력 데이터의 열 수를 의미하는 C, 소수를 의미하는 p, 예외 처리 요구 신호인 TypeD 등이 제어부에 의해서 입력 데이터 수 K에 따라 계산되어 저장되며, 또한 상기 입력 데이터의 행간치환 패턴[T(j)]이 상기 입력 데이터 수 K에 의해 결정되어 미리 저장된다.

<86> 이제 상기 행내치환 패턴[a(j)] 생성부(204)와 행내치환용 기본수열[s(i)] 및 행간치환 패턴[T(j)] 메모리부(212)를 이용한 최종 주소 생성부(207)의 동작을 살펴보기로 한다. 상술한 바와 같이 행내치환 패턴[a(j)] 생성부(202)로부터 출력된 행내 치환 패턴은 레지스터부(200) 내부의 행내치환 기본수열[s(i)] 메모리(208)의 리드 어드레스로 인가된다. 이에 따라 행내치환 기본수열[s(i)] 메모리(208)는 상기 행내치환 패턴[a(j)] 생성부(204)로부터 인가되는 리드 어드레스에 의해 지정된 값인 s[a(j)]를 최종 주소 생성부(212)로 인가시키게 된다.

<87> 이때 상기 리드 어드레스에 지정된 행내치환 기본수열[s(i)] 메모리(208)내 해당

어드레스값이 바로 상기 입력 데이터의 최종 행내치환 패턴[$U_j(i)$]값이므로, 최종 주소 생성부(207)는 상기 행내치환용 기본수열[$s(i)$] 메모리부(208)로부터 리드되는 $s[a(j)]$ 값 및 상기 행간치환 패턴[$T(j)$] 메모리부(212)로부터 리드되는 행간 치환 패턴을 입력 받아 데이터 저장부(214)의 인터리빙된 출력 데이터 리드 어드레스값으로 제공한다.

<88> 이에 따라 상기 데이터 저장부(214)내 상기 리드 어드레스 영역에 저장된 입력 데이터가 리드되어 인터리빙된 출력 데이터로 출력되게 된다.

<89> 따라서 본 발명의 실시 예에서는 입력 데이터의 행간치환시나 행내치환시에 재배열 되는 입력 데이터를 계속 별도의 메모리에 다시 저장하지 않고 순차적으로 저장된 데이터 저장부로부터 최종 행간 또는 행내 치환된 리드 어드레스를 연산하도록 하여 상기 리드 어드레스 해당하는 데이터가 출력되도록 함으로써, 인터리빙된 데이터를 별도로 저장한 메모리로부터 데이터를 순차적으로 리드해내는 종래 인터리버의 기능을 동일하게 수행하는 것이 가능하게 된다.

【발명의 효과】

<90> 상술한 바와 같이 본 발명은 터보부호화기의 내부 인터리버에서의 인터리빙 수행에 있어서, 입력 데이터의 행간치환시나 행내치환시에 재배열 되는 입력 데이터를 별도의 메모리에 다시 저장하지 않고 순차적으로 저장된 입력 데이터 메모리로부터 인터리빙 수행에 따라 최종 행간 또는 행내 치환된 리드 어드레스를 연산하여 상기 입력 데이터 메모리로부터 상기 리드 어드레스에 해당하는 인터리빙된 데이터가 출력되도록 함으로써,

메모리의 낭비를 방지하고, 인터리버 회로의 구성이 간단해지도록 하며, 또한 인터리빙 수행에 따른 터보부호화기의 부하를 줄일 수 있게 되는 이점이 있다.

【특허청구범위】**【청구항 1】**

UMTS시스템내 터보부호화기의 내부 인터리버에 있어서,

인터리버 동작 조건을 설정하기 위한 각종 변수 데이터를 갱신 저장하는 레지스터부와,

상기 레지스터부로부터 조건을 입력받아 시스템 동작에 필요한 제어신호를 생성하는 제어부와,

상기 제어부에서 생성된 제어 신호에 따라 상기 레지스터부에서 제공되는 행내치환 패턴[T(j)], 행내치환 패턴 증가분 배열[incr(i)], 행내치환용 기본수열[s(i)]을 이용하여 최종 인터리빙된 주소를 생성하는 주소 계산부와,

상기 터보 부호화기로 입력되는 입력 데이터를 순차적으로 저장해 두었다가 상기 주소 계산부에서 생성된 주소에 해당하는 데이터를 출력하는 데이터 저장부로 구성됨을 특징으로 하는 UMTS 시스템내 터보 부호화기의 내부 인터리버.

【청구항 2】

제1항에 있어서, 상기 주소 계산부는,

상기 제어부에서 생성된 제어신호에 따라 상기 레지스터부에서 제공되는 상기 행내치환 패턴 증가분 배열[incr(j)]값을 이용하여 행내치환 패턴 데이터열을 계산하는 행내치환 패턴[a(j)] 생성부와,

상기 행내치환 패턴 생성부가 행내치환 패턴 계산시에 중간 데이터를 저장하는 행내치환 패턴 저장 배열부와,

상기 레지스터부로부터의 행간치환 패턴 해당값과 상기 행내치환 패턴 생성부로부터 생성된 행내치환 패턴값에 해당하는 상기 행내치환용 기본수열의 해당값을 취하여 최종 인터리빙된 데이터의 어드레스를 계산하는 최종 주소 생성부로 이루어짐을 특징으로 하는 UMTS 시스템내 터보 부호화기의 내부 인터리버.

【청구항 3】

제1항에 있어서, 상기 레지스터부는,

인터리빙을 위한 상기 입력 데이터의 행간/행내치환 패턴의 계산을 위한 각종 변수를 갱신 저장하며, 이들 변수 데이터를 상기 행내치환 패턴 생성부에 인가하여 인터리빙된 최종 행내 치환 패턴 생성을 위한 행내치환 패턴이 생성되도록 함을 특징으로 하는 UMTS 시스템내 터보 부호화기의 내부 인터리버.

【청구항 4】

제3항에 있어서, 상기 레지스터부는,

상기 행간치환 패턴 $[T(j)]$, 행내치환 패턴 증가분 배열 $[incr(j)]$, 행내치환용 기본수열 $[s(i)]$ 값을 계산하기 위한 변수 데이터로, 입력 데이터 수(K), 소수(μ , p), 입력 데이터의 행 수(R), 입력 데이터의 열 수(C), 예외 처리 요구 신호($TypeD$) 값을 갱신 저장함을 특징으로 하는 UMTS 시스템내 터보 부호화기의 내부 인터리버.

【청구항 5】

제2항에 있어서, 상기 행내치환 패턴 생성부는,

상기 최종 행내 치환 패턴[$U_j(i)$]을 계산하는 과정에서 행간 치환 수열[$r(j)$]을 구하기 위해 상기 행간치환 패턴[$T(j)$]를 역변환한 행간 역치환 패턴[$TI(j)$]를 이용하여 연산을 간략화시킴을 특징으로 하는 UMTS 시스템내 터보 부호화기의 내부 인터리버.

【청구항 6】

제2항에 있어서, 상기 행내치환 패턴 생성부는,

상기 행내치환 패턴 저장부로부터 이전 행내치환 패턴을 읽어 상기 행내치환 패턴 증가분[$incr(j)$]을 가산한 제1가산치를 출력하는 제1가산기와,

상기 제1가산기에서 출력되는 제1가산치에서 제산 계수 $-(p-1)$ 을 가산한 제2가산치를 출력시키는 제2가산기와,

상기 제1가산기와 제2가산기로부터의 두 출력값 중 하나의 값을 선택적으로 출력시키는 제1멀티플렉서와,

상기 제2가산기와 상기 제1멀티플렉서간에 연결되어 상기 제2가산기로부터 출력되는 제2가산치가 양의 값인 경우 상기 제2가산치가 행내치환용 기본 수열[$s(i)$]의 주소로서 출력되도록 하고 제2가산치가 음의 값인 경우 제1가산치의 결과가 행내치환용 기본 수열의 주소로서 출력되도록 상기 제1멀티플렉서로 선택 제어신호를 인가시키는 부호 검출기와,

상기 행내치환 패턴 생성부[$a(j)$]의 초기 동작시 미리 설정된 초기값을

출력시키며, 이후 상기 제1멀티플렉서로부터의 출력을 상기 다음 행내치환 패턴으로서 상기 행내치환 패턴 저장 배열부의 리드 어드레스값으로 인가하는 제2멀티플렉서로 이루어짐을 특징으로 하는 UMTS 시스템내 터보 부호화기의 내부 인터리버.

【청구항 7】

제1항에 있어서, 상기 증가분 배열[incr(j)] 값은,

하기 [수학식 12]에서와 같이 계산되어짐을 특징으로 하는 UMTS 시스템내 터보 부호화기의 내부 인터리버.

【수학식 12】

$$\text{incr}(j) = r\{\text{TI}(j)\} \bmod (p-1)$$

여기서 ,

incr(j): 증가분,

TI(j): 행간 역치환 패턴,

p: 제산 계수,

r(j): 치환후 소수 수열.

【청구항 8】

제6항에 있어서, 상기 행내치환 패턴 저장 배열부는,

상기 제2멀티플렉서로부터 출력되는 한 열의 리드 어드레스값을 순차적으로 저장하며, 이전 저장된 리드 어드레스값을 상기 제1가산기로 피드백 인가시킴을 특징으로 하는

UMTS 시스템내 터보 부호화기의 내부 인터리버.

【청구항 9】

터보부호화기 내부 인터리버에서의 인터리빙 수행 방법에 있어서,

상기 터보부호화로 입력되는 데이터를 상기 입력 데이터 메모리에 순차적으로 저장시키는 과정과,

상기 입력 데이터 수에 따른 행간치환 패턴[$T(j)$]에 따라 상기 입력 데이터의 행내 어드레스를 변환시키는 과정과,

상기 행변환된 어드레스값을 이용하여 행간치환 패턴[$a(j)$] 생성을 위한 증가분[$incr(j)$]값을 계산하는 과정과,

상기 증가분 값과 상기 행내치환 패턴 증가분 배열[$incr(j)$]에 저장된 이전의 행내 치환 패턴값을 이용하여 새로운 행내치환 패턴을 구하는 과정과,

상기 행내치환 패턴을 이용하여 상기 입력 데이터의 행내 어드레스를 변환시키기 위한 행내치환용 기본수열[$s(i)$]의 리드 어드레스를 구하는 과정과,

상기 행내치환용 기본수열 메모리의 출력에 따라 최종 행내 치환 패턴을 계산하는 과정과,

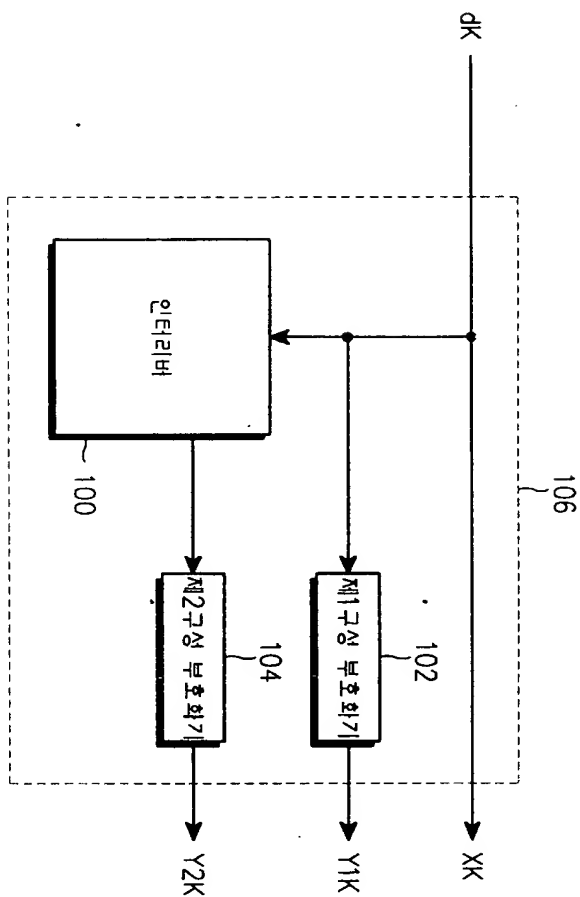
상기 최종 행내 치환 패턴[$U_j(i)$] 및 상기 행간치환 패턴[$T(j)$]을 이용하여 최종 인터리빙된 주소를 계산하는 과정과,

상기 입력 데이터 메모리로부터 상기 행간/행내 치환된 최종 인터리빙된 어드레스

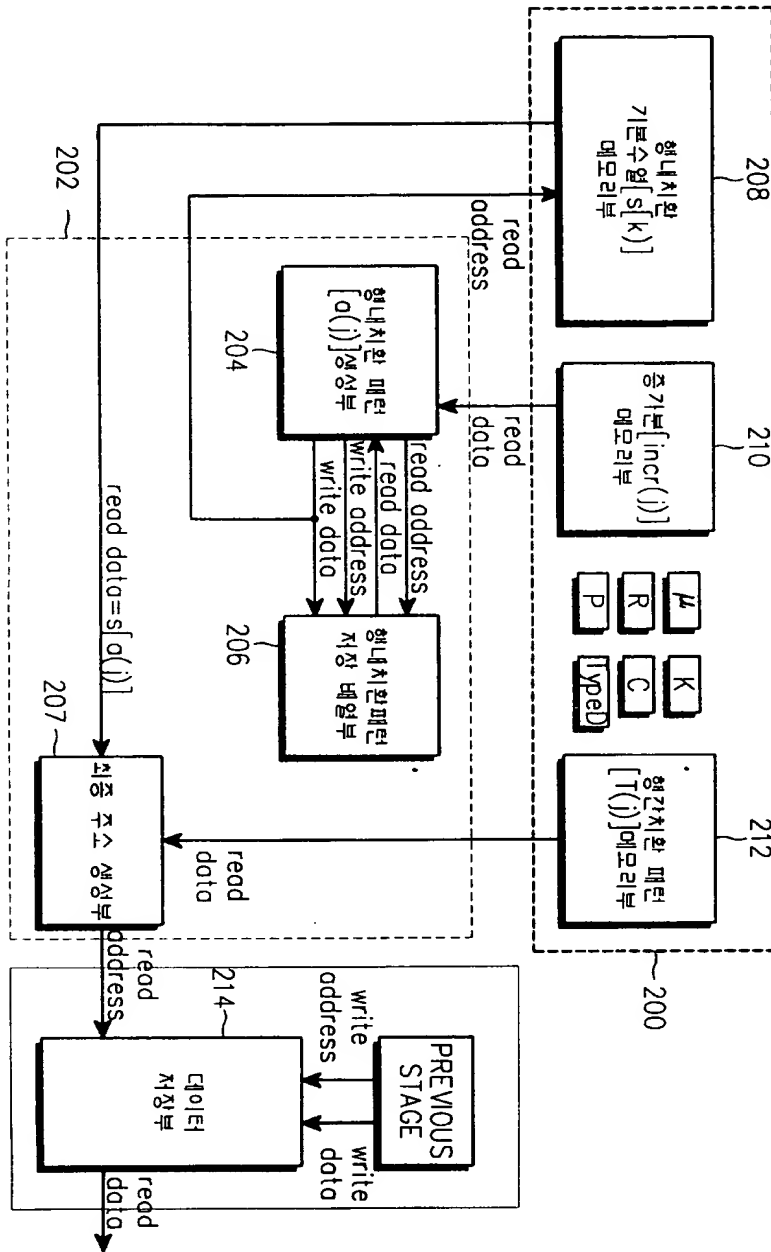
에 해당하는 데이터를 순차적으로 출력하여 입력 데이터에 대한 인터리빙을 수행하는 과정으로 구성됨을 특징으로 하는 터보부호화기 내부 인터리버에서의 인터리빙 수행 방법.

【도면】

【도 1】



【도 2】



【도 3】

